Efficient Time Slot Assignment Algorithms for TDM Hierarchical and Nonhierarchical Switching Systems

Kwan L. Yeung

Abstract—**Two efficient time slot assignment algorithms, called the two-phase algorithm for the nonhierarchical and the three-phase algorithm for the hierarchical time-division multiplex (TDM) switching systems, are proposed. The simple idea behind these two algorithms is to schedule the traffic on the critical lines/trunks of a traffic matrix first. The time complexities of these two algorithms are found to be $O(LN^2)$ and $O(LM^2)$, where L is the frame length, N is the switch size, and M is the number of input/output users connected to a hierarchical TDM switch. Unlike conventional algorithms, they are fast, iterative and simple for hardware implementation. Since no backtracking is used, pipelined packet transmission and packet scheduling can be performed for reducing the scheduling complexity of a transmission matrix to $O(N^2)$ and $O(M^2)$, respectively. Extensive simulations reveal that the two proposed algorithms give close-to-optimal performance under various traffic conditions.**

Index Terms—**Heuristic, scheduling algorithm, TDM switching system, time slot assignment.**

I. Introduction

TIME-DIVISION multiplex (TDM) switching has been widely employed in terrestrial and satellite communication networks to concentrate traffic from low bandwidth sources onto high bandwidth lines [1]–[3]. A TDM switching system can be either nonhierarchical or hierarchical. A nonhierarchical TDM switch is shown in Fig. 1. The switching operation is made up of frames and each frame is divided into time slots. Each time slot can accommodate one packet. A switch configuration is an interconnection pattern of the switch such that at most one packet can be transmitted by an input and one packet can be received by an output. An $N \times N$ switch has a switch configuration that permits up to N packets to be transmitted in a conflict-free manner from inputs to outputs in each time slot. A switching conflict occurs if two or more packets are transmitted to the same output in the same time slot.

The TDM hierarchical switching system was first proposed in [4]. Subsequent studies can be found, e.g., in [5] and [6]. The TDM hierarchical switching system has a three-stage switching structure with M input users and M output users as shown in Fig. 6. The first and the third stages consist of multiplexers and demultiplexers. The second stage has the same structure as that of the nonhierarchical switch. If the number of the inputs and outputs for all multiplexers and demultiplexers are equal, a hierarchical TDM switching system degenerates into a nonhierarchical one.

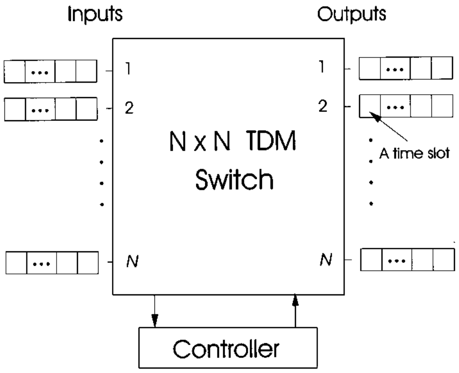

Fig. 1. An $N \times N$ nonhierarchical TDM switching system.

A traffic matrix is used to represent the total amount of traffic from all inputs to all outputs of a switch (a formal definition will be defined later). For a given traffic matrix, the time slot assignment (TSA) problem is to find a conflict-free assignment of packets to slots such that the frame length, i.e., the number of time slots required for switching the traffic from switch inputs to outputs, is minimized. An optimal TSA is an assignment that has the minimum frame length over all possible conflict-free assignments.

Time slot assignment as well as traffic scheduling problems have also been broadly studied in other areas, including design of multirate TDM video switch [7] and input-buffered packet switches [8]–[14]. There is also a growing interest in using TSA algorithms to provide quality-of-service guarantee [15], [16].

For both hierarchical and nonhierarchical switching systems, many optimal algorithms with polynomial time complexities have been proposed [4], [1], [2], [17]–[20]. Most of them use a maximum cardinality algorithm known as the system of distinct representatives (SDR) [1], or maximum size matching algorithm. For nonhierarchical TDM switching systems, an algorithm with complexity $O(m^3)$, where m is the number of nonzero entries in the traffic matrix, was proposed in [2]. Using the implementation of the maximum matching algorithm in [21], an algorithm with complexity $O(L_{\min}N^{2.5})$ was reported, where $L_{\min}$ is the frame length of the optimal time slot assignment. For hierarchical TDM switching systems, an algorithm with time complexity of $O(\min(L_{\min}, M^2) \cdot \min(N, \sqrt{M}) \cdot M^2)$ was studied in [18], where M is the total number of input users. In [22], a

Paper approved by A. Pattavina, the Editor for Switching and Architecture Performance of the IEEE Communications Society. Manuscript received May 12, 1998; revised April 9, 1999, May 9, 2000, and June 26, 2000. This work was supported by the CERG, Hong Kong, China, under Grant 9040264. This paper was presented in part at the IEEE GLOBECOM Conference, London, U.K., November 1996.

The author was with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. He is now with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: kyeung@eee.hku.hk).

Publisher Item Identifier S 0090-6778(01)01298-3.

slightly improved algorithm with time complexity $O(M^4)$ was proposed using a more efficient implementation of the maximum size matching algorithm. Algorithms using the neural network approach have also been reported [6]. For high-speed communications networks such as ATM, the complexities of those reported algorithms are too high to be run in real-time and are too complicated to be implemented in hardware. Therefore, a near-optimal algorithm with a much lower time complexity that can be easily implemented in hardware is desirable.

In this paper, two efficient heuristic TSA algorithms, the two-phase algorithm for nonhierarchical and the three-phase algorithm for hierarchical TDM switching systems, are proposed. They are designed based on the idea that the traffic on the critical lines/trunks (to be defined) of a traffic matrix should be scheduled first. The time complexities of the two proposed algorithms are $O(LN^2)$ and $O(LM^2)$, respectively, where L is the frame length, N is the switch size, and M is the number of input users. Since no backtracking is used in the proposed algorithms, scheduled packets for time slot k can be transmitted while the algorithm is scheduling for time slot $k+1$. In other words, packet scheduling and transmission can be operated in a pipelined fashion. Therefore, the complexity for scheduling a transmission matrix is only $O(N^2)$ or $O(M^2)$. The parameter L will not affect the scheduling complexity in each time slot.

Because of the iterative nature of the two proposed algorithms, they are simple for hardware implementation and thus very suitable for high-speed switching networks. Extensive simulations reveal that the two-phase and the three-phase algorithms are very efficient. When switch size N is large, we found that for nonhierarchical switching systems P_F, the probability that the two-phase algorithm failed to generate optimal TSAs, is almost independent of N, and has a very low value of $P_F = 3 \times 10^{-5}$. For hierarchical switching systems, the percentage increase in frame length as a result of nonoptimal TSAs by the three-phase algorithm is only about 0.1%.

This paper is organized as follows. In the next two sections, we focus on the designs of the two-phase and the three-phase algorithms, respectively. For each section, we start with a formal problem formulation and then follow by a detailed algorithm description. The performance of the proposed algorithm is then studied by simulations.

II. Nonhierarchical Switching Systems

A. Problem Formulation

Consider an $N \times N$ time multiplexed switch as shown in Fig. 1. Packets from all inputs can be represented by a traffic matrix $D = [d_{ij}]$, where d_{ij} is the number of packets to be transmitted from input i to output j. Let the packets to be transmitted in time slot k be represented by a transmission matrix $T_k = [t_{ij}^{(k)}]$, where $t_{ij}^{(k)} = 1$ if a packet is scheduled to transmit from input i to output j in the current time slot; otherwise, $t_{ij}^{(k)} = 0$. For a conflict-free transmission, each row and each column of T_k can have at most one 1. For a given traffic matrix D, the time slot assignment is to decompose D into a series of transmission matrices, one for each time slot, or $D = T_1 + T_2 + \cdots + T_L$, where L is the frame length, or the length of a time slot assignment. We further let D_k denote the (remaining) traffic matrix at the end of time slot k, i.e., $D_k = D - \sum_{i=1}^{k} T_i$. An optimal time slot assignment is an assignment that has the minimum length $L_{\min}$ such that $D_{L_{\min}} = 0$.

$$D = \begin{array}{c} \left[\begin{array}{cccc} 1 & 2 & 1 & 0 \\ 2 & 0 & 2 & 1 \\ 2 & 1 & 1 & 2 \\ 0 & 0 & 0 & 3 \end{array}\right] \begin{array}{c} 4 \\ 5 \\ \mathit{6} \\ 3 \end{array} \\ \begin{array}{cccc} 5 & 3 & 4 & \mathit{6} \end{array} \quad \end{array}$$

Fig. 2. An example traffic matrix D for a 4×4 nonhierarchical TDM switch.

Let $R_i = \sum_j d_{ij}$ and $C_j = \sum_i d_{ij}$ be the ith row sum and the jth column sum of the traffic matrix D. The following theorem has been shown in [23].

Theorem 1: The necessary and sufficient number of time slots required to transmit a traffic matrix D is $L_{\min} = \max_{i,j}\{R_i, C_j\}$.

An optimal time slot assignment algorithm is an algorithm that finds an assignment with $L = L_{\min}$ in all cases.

B. Two-Phase TSA Algorithm

Let a *line* be a row or a column in a traffic matrix D. Let a *critical* line be a line which has the maximum traffic. Consider a 4×4 traffic matrix D as shown in Fig. 2. The row and the column sums of D are shown at the right-hand side and the bottom of D, respectively. We can see that row 3 and column 4 are critical lines with (maximum) traffic 6.

Let $R_i^{(k)}$ and $C_j^{(k)}$ be the ith row sum and the jth column sum of the traffic matrix D_k. Further let $L_{\min}^{(k)} = \max_{i,j}\{R_i^{(k)}, C_j^{(k)}\}$. A critical line of matrix D_k (at the beginning of time slot $k+1$) is a line with row sum or column sum equals to $L_{\min}^{(k)}$. From Theorem 1, the following corollary can be easily shown.

Corollary 1: The necessary and sufficient condition for an algorithm to obtain an optimal time slot assignment is that for each time slot, exactly one packet is transmitted from each critical line of the traffic matrix D_k. In other words, $L_{\min}^{(k)} = L_{\min}^{(k-1)} - 1 = L_{\min} - k$ must be satisfied for all $k > 0$.

We can see that it is *not* necessary for an optimal TSA algorithm to find a transmission matrix with maximum cardinality in each time slot. This is, however, the approach used by most of the exsiting optimal TSA algorithms. The algorithms for finding a transmission matrix with maximum cardinality [1] use backtracking and their time complexities tend to be very high. From Corollary 1, we can see that a more efficient optimal algorithm can be designed by only focusing on scheduling packets on the critical lines.

In this paper, we propose an efficient time slot assignment algorithm based on the observation that packets on critical lines should have scheduling priority over those on the noncritical lines. We call it the two-phase algorithm because for scheduling each transmission matrix, the algorithm consists of two phases, where

- phase 1 schedules the packets on critical lines of the traffic matrix;
- phase 2 schedules the packets on the remaining noncritical lines.

The two phases of the two-phase algorithm are both implemented using an iterative algorithm called maximum remaining sum (MRS) [24]. The MRS algorithm is an efficient heuristic algorithm for finding a matrix with maximum cardinality. Unlike the system of distinctive representative [1] algorithm, no backtracking is needed and its time complexity is found to be only $O(N^2)$. During its operation, scheduling priority is given to a nonzero traffic matrix entry (i, j), where the total number of zero entries in row i and column j is maximum among all other rows and columns. The idea is to give priority to the entry which has the least assignment/scheduling choices (becasue it has the largest number of zero entries, where a zero entry means there is no traffic there for you to schedule).

The detailed operations of the two-phase algorithm are described by the following pseudocodes and the MRS algorithm is also summarized below.

```
Two-Phase TSA Algorithm
Inputs:   $D = [d_{ij}]$
Outputs:  $T_1, T_2, \ldots, T_L$
----------------------------------------------------------
1. $k = 1$;
2. Find all critical lines in $D; D' = D$;
   Set all entries of $D'$ to 0 except those on the critical lines;
3. Run MRS algorithm with input $D'$ and $T_k = [0]$;
4. $D' = D$; for all $t_{ij}^{(k)} = 1$, set the corresponding lines in
   $D'$ to 0;
5. Run MRS algorithm with input $D'$ and $T_k$;
6. $D = D - T_k$;
7. If $D = [0], L = k$; EXIT;
   Else $k = k + 1$; goto step 2.
```

```
MRS Algorithm
Inputs:   $D'$ and $T_k$
Outputs:  $T_k$
----------------------------------------------------------
1. Find $r_i$ and $c_i$, the number of nonzero entries for each line
   of $D'$;
2. Find the line with minimum number of nonzero entries.
   If the line is a row, denote it by $p$ and goto step 3.
   If the line is a column, denote it by $q$ and goto step 4;
3. Find column $q$ such that $c_q = \min_i\{c_i \mid d_{pi} > 0\}$;
4. Find row $p$ such that $r_p = \min_i\{r_i \mid d_{pi} > 0\}$;
5. $t_{pq}^{(k)} = 1$ and set all entries in row $p$ and column $q$ of $D'$ to 0;
   $r_i = r_i - t_{iq}$ and $c_i = c_i - t_{pi}$ for all $i$'s.
6. Goto step 2 until no packet can be scheduled into $T_k$.
```

Steps 2 and 3 of the two-phase algorithm correspond to phase 1, and steps 4 and 5 correspond to phase 2. Steps 2–7 execute L times for scheduling L transmission matrices, one for each time slot. In each scheduling iteration, the MRS sub-routine is executed twice (one for each phase) and the time complexity for MRS is $O(N^2)$ [24]. Therefore, for a given traffic matrix, the overall time complexity of the two-phase algorithm is $O(LN^2)$.

On the other hand, the two-phase algorithm does not perform any backtracking. That means once a packet is scheduled/assigned to a transmission matrix, its assignment will not be changed by the subsequent scheduling assignments. Therefore, packet transmission and packet scheduling can be operated in a pipelined fashion. That is after each scheduling iteration, say the kth iteration, the scheduled packets in transmission matrix T_k can be immediately transmitted while the next iteration for scheduling transmission matrix T_{k+1} is being carried out. Because scheduling and transmission are now overlapped in time, the complexity for scheduling a single transmission matrix is only $O(N^2)$. It should be noticed that this pipelined operation *cannot* be used by algorithms using backtracking. For a backtracking-typed algorithm, no packet can be transmitted until the transmission matrices for the whole traffic matrix are obtained.

C. Some Implementation Consideration

Unlike the maximum size matching algorithm [1], [21], the two-phase algorithm is a fast and iterative heurtistic. It is very simple to implement in hardware. The complexity of the two-phase algorithm is dominated by the function MRS. A simple special vector processor [24] can be used for updating the traffic matrix, the row sum and the column sum in step 6 of MRS. Then step 6 can be carried out in a constant number of steps which is independent of switch size N. Also simple combinatorial logic can be used to perform the vector comparison in steps 2–4. The time complexity of the MRS algorithm can then be reduced to $O(N)$ and the time complexity of the two-phase algorithm for scheduling a transmission matrix also becomes $O(N)$. To further speed up the algorithm, one can follow the approach of using hardware implementation for an iterative algorithm as reported in [11] and [9].

D. An Example

To illustrate the two-phase TSA algorithm, consider a 4 × 4 TDM switch with a traffic matrix given by

$$D = \begin{array}{c} \begin{bmatrix}1&2&1&0\\2&0&2&1\\2&1&1&2\\0&0&0&3\end{bmatrix} \begin{array}{c}4\\5\\6\\3\end{array} \\ \begin{array}{cccc}5&3&4&6.\end{array}\end{array}$$

The row and column sums of the traffic matrix are shown at the right-hand side and the bottom of D, respectively. Row 3 and column 4 are critical lines. Following step 2, a running matrix D' is constructed.

$$D' = \begin{array}{c} \begin{bmatrix}0&0&0&0\\0&0&0&1\\2&1&1&2\\0&0&0&3\end{bmatrix} \begin{array}{c}-\\-\\4\\-\end{array} \\ \begin{array}{cccc}-&-&-&3.\end{array}\end{array}$$

Then we excecute the MRS algorithm in step 3. The number of nonzero entries of the critical lines are found by step 1 of MRS and are shown next to row 3 and column 4. Step 2 of MRS identifies column 4 is the line with the minimum nonzero entries. Then by step 4, row 3 is chosen and a packet from input 3 to output 4 is scheduled

$$T_1 = \begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&1\\0&0&0&0\end{bmatrix}.$$

Since no more packet in D' can be scheduled, exit MRS algorithm and return to the two-phase algorithm at step 4. We obtain

$$D' = \begin{array}{cc} \begin{bmatrix} 1 & 2 & 1 & 0 \\ 2 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{matrix} 3 \\ 2 \\ - \\ - \end{matrix} \\ \begin{matrix} 2 & 1 & 2 & - \end{matrix}. & \end{array}$$

The number of nonzero entries of each row and column are again shown next to their respective lines. Next, we call the MRS algorithm again at step 5. Column 2 of D' is chosen since it has the minimum number of nonzero entries. Similarly row 1 is chosen and a packet from input 1 to output 2 is scheduled. The resulting transmission matrix is

$$T_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Follow the MRS algorithm, the updated matrix D' is obtained and then we proceed to the second iteration of MRS.

$$D' = \begin{array}{cc} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \begin{matrix} - \\ 2 \\ - \\ - \end{matrix} \\ \begin{matrix} 1 & - & 1 & - \end{matrix}. & \end{array}$$

Finally, the transmission matrix for the first time slot is obtained

$$T_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The traffic matrix D is then updated in step 6 of the two-phase algorithm.

$$D = \begin{array}{cc} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 2 & 1 \\ 2 & 1 & 1 & 1 \\ 0 & 0 & 0 & 3 \end{bmatrix} & \begin{matrix} 3 \\ 4 \\ 5 \\ 3 \end{matrix} \\ \begin{matrix} 4 & 2 & 4 & 5 \end{matrix}. & \end{array}$$

Skipping the remaining steps, the transmission matrices for the subsequent time slots are found to be

$$T_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T_4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_5 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_6 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Note that the total number of time slots required for this example is 6 and is equal to $L_{\min}$, the maximum line traffic of the original D. Therefore, from Theorem 1, the resulting time slot assignment is optimal.

E. Performance Evaluation

In this section, the performance of the two-phase TSA algorithm is studied by simulations. To examine the effect of giving priority to critical lines, two simplified versions of the two-phase algorithm are constructed. We call them the phase-1-only algorithm and the phase-2-only algorithm. The phase-1-only algorithm consists of steps 1–3, 6, and 7 of the original two-phase algorithm, while the phase-2-only algorithm consists of steps 1 and 4–7. We can see that the phase-1-only agorithm can take advantage of scheduling critical lines first, but not the throughput maximization offered by the second phase. And the vice versa is true for the phase-2-only algorithm.

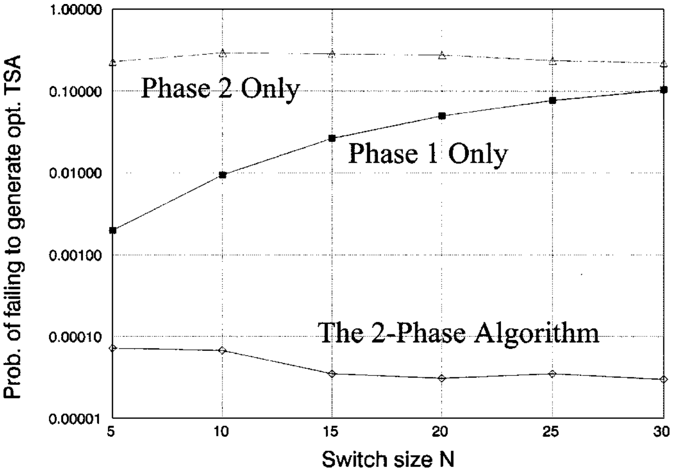

Fig. 3. Probability that the proposed algorithms failed to generate optimal TSA versus switch size $N; n = 4$.

Let d_{ij} the number of packets from input i to output j be a random integer uniformly distributed between 0 and n. The input traffic load increases as n increases. Let $L - L_{\min}$ be the difference in frame length generated by a heuristic algorithm and that by an optimal algorithm. $L - L_{\min} = 0$ if an optimal assignment is obtained using a heuristic algorithm. Otherwise, $L - L_{\min} > 0$. Fig. 3 shows in logarithmic scale that P_F, the probability that a heuristic algorithm failed to generate optimal TSA (i.e., $L - L_{\min} > 0$), versus switch size N for $n = 4$. Each point of the simulation results shown in the figure is obtained by averaging over one million traffic matrices.

For the phase-1-only algorithm, the probability of generating suboptimal TSA P_F increases almost linearly with switch size.[1] For the phase-2-only algorithm, P_F is the highest and it has a value between 0.22–0.29. It shows that scheduling the traffic on the critical lines first is very important. For the two-phase algorithm, P_F is the lowest and when N is large, P_F is almost independent of the switch size. When switch size $N = 30, P_F$ is found to be 0.10 for the phase-1-only algorithm, 0.22 for the phase-2-only algorithm, and 3×10^{-5} for the two-phase algorithm. This represents a three order of magnitude improvement over the phase-1-only and phase-2-only algorithms. This significant performance improvement comes from the facts that 1) scheduling priority is given to the critical lines (using phase 1), and 2) the number of packets sent in each slot is maximized (using phase 2).

Next, we concentrate on the quality of the time slot assignments given that the suboptimal assignments are obtained, i.e., $L - L_{\min} > 0$. We found that when the two-phase algorithm is used, the frame length difference $L - L_{\min}$ is at most equal

[1] It is easier to see this trend when y-axis is in linear scale.

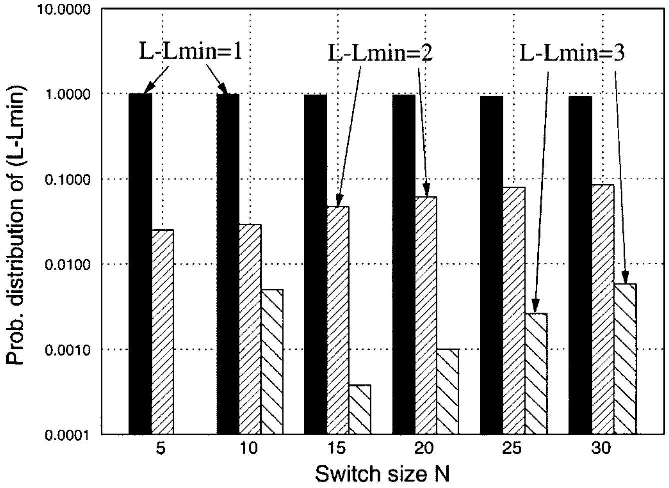

Fig. 4. Probability distribution of the frame length difference between the phase-1-only algorithm and the optimal; $n = 4$.

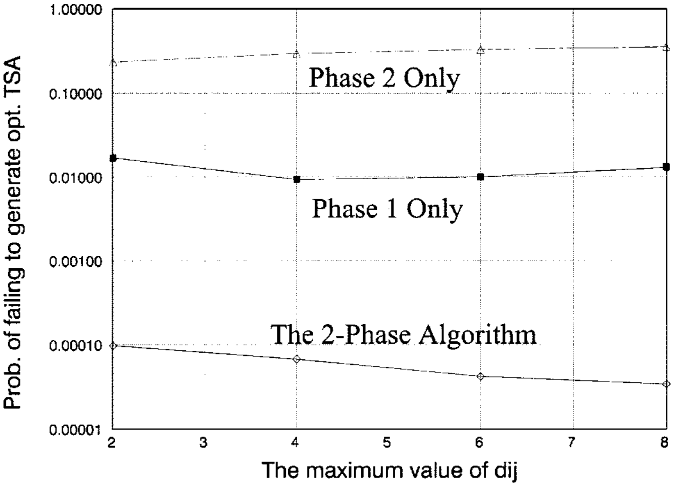

Fig. 5. Probability that the proposed algorithm failed to generate optimal TSA versus $n; N = 10$.

to 1. That means all suboptimal assignments are just *one* time slot longer than the optimal. For the phase-1-only algorithm, the frame length difference $L - L_{\min}$ spreads out to values 1, 2 and 3. Fig. 4 shows their respective distributions against the switch size. It can be seen that the majority of the suboptimal cases have $L - L_{\min} = 1$. As switch size N increases, the probability of $L - L_{\min} = 1$ is slightly reduced and the probabilities of $L - L_{\min} = 2$ and 3 are slightly increased. Similar results (not shown) are obtained for the phase-2-only algorithm.

Fig. 5 shows P_F versus n, the maximum value for the traffic matrix entry d_{ij}, with a switch size of 10. As n increases, the input traffic load increases. For the phase-1-only and the phase-2-only algorithms, P_F is around 0.3 and 0.01, respectively. For the two-phase algorithm, it is interesting to notice that P_F decreases with n. This is because the line sums of the traffic matrix spread out to a much wider range as n increases. This causes the number of critical lines (lines with the maximum traffic) in a traffic matrix to decrease. From Corollary 1, a TSA algorithm fails to generate an optimal TSA if it fails to schedule a critical line. If the number of critical lines decreases, the chance of scheduling failure (i.e., fail to transmit exactly one packet for each critical line in each time slot) is reduced. An important implication from this result is that the two-phase algorithm will give a better performance under nonuniformly distributed traffic situations than the uniformly distributed traffic we examined here.

Next, we focus on the worst-case scenario that all $2N$ lines in an $N \times N$ traffic matrix are critical. To generate such an matrix, we simply set all entrires of a traffic matrix equal to a constant c, where $c = 1$ is used. The switch size (or, matrix size) N is varied from 2 to 100 one by one. We found that when $N = 42, 49, 54, 56, 66, 68, 70, 77, 80, 81, 84, 85, 91$, and 93 (14 cases out of 99), the two-phase algorithm generates suboptimal time slot assignments. But in all such suboptimal cases, again only one extra time slot is required. We then vary the value of the constant c from 1 to 10 for all 14 suboptimal cases above, we found that still only one extra time slot is required in each case. That implies when all entries of the traffic maxtrix are equal to c, the value of c does not affect or has little impact on the quality of the time slot assignments generated. From the above worst-case scenerio, we can see that the two-phase algorithm still generates a close-to-optimal performance but the performance improvement is reduced as expected.

It should also be noticed that the two-phase algorithm is designed and evaluated for minimizing the frame length in this paper. For packet switching systems [24], [8], [14], [13] where packets arrive at inputs on a slot-by-slot basis, the two-phase algorithm can still be applied with the modification that the traffic matrix is updated in each time slot.

III. Hierarchical Switching Systems

A. Problem Formulation

In this section, we focus on a hierarchical TDM switching system. A TDM hierarchical switching system has a three-stage switching structure as shown in Fig. 6. The first stage consists of f multiplexers and a total of $M = \sum_{i=1}^{f} p_i$ input users, where multiplexer i concentrates p_i input users to k_i lines. (Thus $p_i \geq k_i$.) A total of $N = \sum_{i=1}^{f} k_i$ concentrated input lines are then connected to the $N \times N$ nonblocking TDM switch in the middle stage. The N output lines of the TDM switch are then grouped into g sets, where set i with h_i lines is connected to the ith demultiplexer in the third stage. The number of output lines/users of demultiplexer i is q_i (where $q_i \geq h_i$) and $\sum_{i=1}^{g} q_i = M$. When $p_i = k_i$ and $h_i = q_i$ for all i's, the hierarchical switching system degenerates into a nonhierarchical system we studied in the previous section.

Packets from all M input users are represented by a traffic matrix $B = [b_{ij}]_{M \times M}$, where b_{ij} is the number of packets to be transmitted from input user i to output user j. Let $S = \sum_i \sum_j b_{ij}$ be the total number of packets in B. Let the M input users be grouped into f sets and each set corresponds to a multiplexer. We denote the set of users at multiplexer i as I_i. Similarly, we group the output users into g sets and denote each set by O_j. We call each set I_i an input *trunk* and each set O_j an output *trunk*. Let $R_i = \sum_{j=1}^{M} b_{ij}$ and $C_j = \sum_{i=1}^{M} b_{ij}$ denote the total traffic in row i and column j of matrix B, respectively. Let $U_i = \sum_{j \in I_i} R_j$ and $V_j = \sum_{i \in O_j} C_i$ denote the total traffic in (row) trunk i and (column) trunk j, respectively.

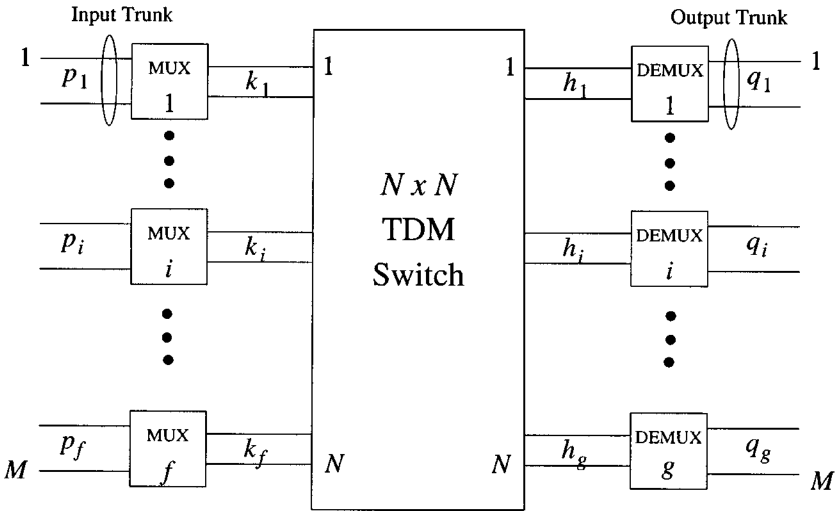

Fig. 6. A three-stage hierarchical TDM switching system.

For a given traffic matrix, the time slot assignment problem is to find a way to decompose traffic matrix B into a series of zero-one transmission matrices, or $B = T_1, T_2, \ldots, T_L$, where frame length L is minimized. Unlike a nonhierarchcial system, the following *two* conditions must be met for a valid transmission matrix $T_k = [t_{ij}^{(k)}]$: i) at most one 1 in each row or each column of T_k, and ii) at most k_i 1's in trunk I_i and h_j 1's in trunk O_j.

An optimal time slot assignment is the assignment which has the minimum frame length $L_{\min}$. It was shown [4], [19] that a time slot assignment with length $L_{\min}$ exists if and only if the following conditions are satisfied:

$$\begin{aligned} &S \le NL_{\min} \\ &R_i \le L_{\min}, \quad \text{for } 1 \le i \le M \\ &C_j \le L_{\min}, \quad \text{for } 1 \le j \le M \\ &U_r \le k_r L_{\min}, \quad \text{for } 1 \le r \le f \\ &V_s \le h_s L_{\min}, \quad \text{for } 1 \le s \le g. \end{aligned}$$

The following theorem has been proved [4], [19].

Theorem 2: The necessary and sufficient number of time slots to transmit traffic matrix B is

$$L_{\min} = \max\left\{ \lceil S/N \rceil, \max_i\{R_i\}, \max_j\{C_j\}, \max_r\{\lceil U_r/k_r \rceil\}, \max_s\{\lceil V_s/h_s \rceil\} \right\}. \quad (1)$$

B. Three-Phase TSA Algorithm

Let a *critical* trunk be a trunk with $\lceil U_r/k_r \rceil = L_{\min}$ or $\lceil V_s/h_s \rceil = L_{\min}$. Note that a critical line may or may not reside in a critical trunk. Unlike that for nonhierarchical switching systems, a critical line or a critical trunk may not exist. This happens when the maximum term on the left-hand side of (1) is $\lceil S/N \rceil$, i.e., $L_{\min} = \lceil S/N \rceil$. From Theorem 2, we found that if the following conditions are not satisfied, an optimal time slot assignment cannot be obtained.

Corollary 2: The necessary and sufficient conditions for an algorithm to obtain an optimal time slot assignment are, for each time slot, as follows:

1) exactly one packet is transmitted from each critical line (if any) of traffic matrix B;
2) the number of packets transmitted from each critical trunk (if any) is sufficient to make it become noncritical, i.e., $\lceil U_r/k_r \rceil$(after) $=$ $\lceil U_r/k_r \rceil$(before) $-$ 1 or $\lceil V_s/h_s \rceil$(after) $=$ $\lceil V_s/h_s \rceil$(before) $-$ 1, where $\lceil \cdot \rceil$(before) and $\lceil \cdot \rceil$(after) denote the value of $\lceil \cdot \rceil$ before and after the time slot assignment in a time slot.

To design an efficient TSA algorithm for hierarchical switching systems, we concentrate on scheduling the traffic on critical lines and critical trunks. For scheduling a critical line, the algorithm needs to decide which packet in the given critical line to transmit. For scheduling a critical trunk, the algorithm has more flexibility as it needs to decide which packet from *which* line (in the critical trunk under consideration) to transmit. An efficient algorithm should give priority to the assignment with higher assignment difficulty, or less assignment flexibility. Therefore, scheduling for critical lines should have a higher priority over the scheduling for critical trunks, and scheduling for critical trunks should have a higher priority over the scheduling for the remaining traffic.

Following this approach, a new algorithm called the three-phase algorithm is proposed. It consists of three phases for scheduling a transmission matrix in each time slot, where phase 1 has the highest priority and phase 3 has the lowest priority. To have a easy-to-follow presentation of each phase, we try to use less symbols and notations below.

- *Phase 1 focuses on scheduling traffic on the critical lines while giving priority to lines inside a critical trunk.* For all critical lines in traffic matrix B, find the critical line with the minimum number of nonzero entries (i.e., has the minimum number of assignment choices).

  1) If the line is a row, denote it by p and the associated row trunk (i.e., the trunk where line p resides in) by I_i. Find a column line according to the following order and denote it by q and the associated column

trunk by O_j. The number of 1's in trunks I_i and O_j of the transmission matrix must be less than k_i and h_j, respectively.

   a) Find a critical column q such that it has the minimum number of nonzero entries among all critical columns with $b_{pq} > 0$.

   b) Find a column q such that it has the minimum number of nonzero entries among all columns within critical column trunks and with $b_{pq} > 0$.

   c) Find a column q such that it has the minimum number of nonzero entries among all columns with $b_{pq} > 0$.

2) If the line is a column, denote it by q and the associated column trunk by O_j. Find a row line according to the following order and denote it by p and the associated row trunk by I_i. The number of 1's in trunks I_i and O_j of the transmission matrix must be less than k_i and h_j, respectively.

   a) Find a critical row p such that it has the minimum number of nonzero entries among all critical rows with $b_{pq} > 0$.

   b) Find a row p such that it has the minimum number of nonzero entries among all rows within critical row trunks and with $b_{pq} > 0$.

   c) Find a row q such that it has the minimum number of nonzero entries among all rows with $b_{pq} > 0$.

3) Schedule a packet to be sent from input user p to output user q by setting $t_{pq}^{(k)} = 1$. Repeat phase 1 until no more critical lines are found.

- *Phase 2 focuses on scheduling traffic on the critical trunks.* It should be noticed that after performing phase 1, some critical trunks may now become noncritical.[2] Among all critical trunks, find the line with the minimum number of nonzero entries.

   1) If the line is a row, denote it by p and the associated row trunk by I_i. Find a column line according to the following order and denote it by q and the associated column trunk by O_j. The number of 1's in trunks I_i and O_j of the transmission matrix must be less than k_i and h_j, respectively.

      a) Find a column q such that it has the minimum number of nonzero entries among all columns within critical coulumn trunks and with $b_{pq} > 0$.

      b) Find a column q such that it has the minimum number of nonzero entries among all columns with $b_{pq} > 0$.

   2) If the line is a column, denote it by q and the associated column trunk by O_j. Find a row line according to the following order and denote it by p and the associated row trunk by I_i. The number of 1's in trunks I_i and O_j of the transmission matrix must be less than k_i and h_j, respectively.

      a) Find a row p such that it has the minimum number of nonzero entries among all rows within critical row trunks and with $b_{pq} > 0$.

      b) Find a row p such that it has the minimum number of nonzero entries among all rows with $b_{pq} > 0$.

   3) Schedule a packet to be sent from input user p to output user q by setting $t_{pq}^{(k)} = 1$. Repeat phase 2 until no more critical trunks are found.

- *Phase 3 schedules the remaining traffic in the matrix in order to maximize the number of packets to be transmitted in a time slot, i.e., throughput.* Find the line with the minimum number of nonzero entries.

   1) If the line is a row, denote it by p and the associated row trunk by I_i. Find a column line q and denote the associated column trunk by O_j. Column q is found such that

      a) it has the minimum number of nonzero entries among all columns with $b_{pq} > 0$;

      b) the number of 1's in trunks I_i and O_j of the transmission matrix must be less than k_i and h_j, respectively.

   2) If the line is a column, denote it by q and the associated column trunk by O_j. Find a row line p and denote the associated row trunk by I_i. Row p is found such that

      a) it has the minimum number of nonzero entries among all rows with $b_{pq} > 0$;

      b) the number of 1's in trunks I_i and O_j of the transmission matrix must be less than k_i and h_j, respectively.

   3) Schedule a packet to be sent from input user p to output user q by setting $t_{pq}^{(k)} = 1$. Repeat Phase 3 until no more packets can be added to the transmission matrix.

Similar to the two-phase algorithm, the overall time complexity of the three-phase algorithm is found to be $O(LM^2)$. For scheduling a single time slot transmission, the time complexity is $O(M^2)$ since pipelined operation of scheduling and transmission can be used. Like the two-phase algorithm, the three-phase algorithm can also be easily implemented in hardware for faster operations. Its circuit complexity will be higher than that of the two-phase algorithm because three levels of scheduling priority are required.

C. Performance Evaluation

Consider a symmetric hierarchical TDM switching system with $p_i = q_i = 4$ and $k_i = h_i = 2$. Then for an $M \times M$ hierarchical switching system, the $N \times N$ TDM switch in the middle has a size of $N = M/2$. Let b_{ij} the number of packets from input source i to output source j be a random integer uniformly distributed between 0–4. Fig. 7 shows P_F, the probability that the three-phase failed to generate optimal TSA, and

[2] In case a critical line resides in a critical trunk, if a packet on this critical line can be successfully transmitted in phase 1, the corresponding critical line becomes noncritical but the the corresponding critical trunk is not necessarily to become noncritical. This is because of the ceiling functions on expressions U_r/k_r and V_s/h_s in (1).

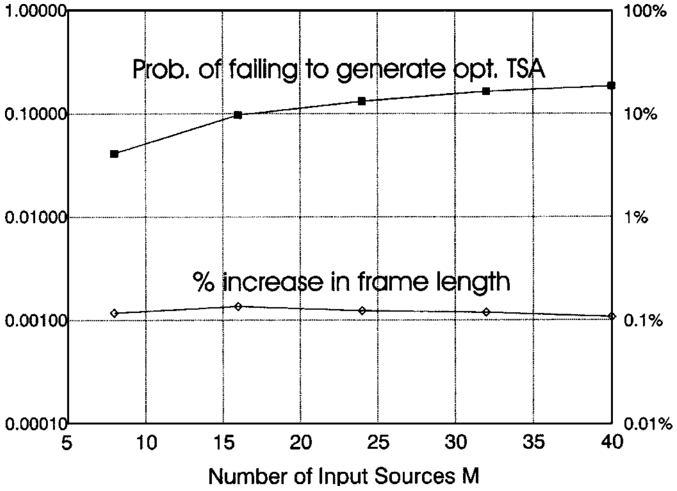

Fig. 7. Performance of the three-phase algorithm.

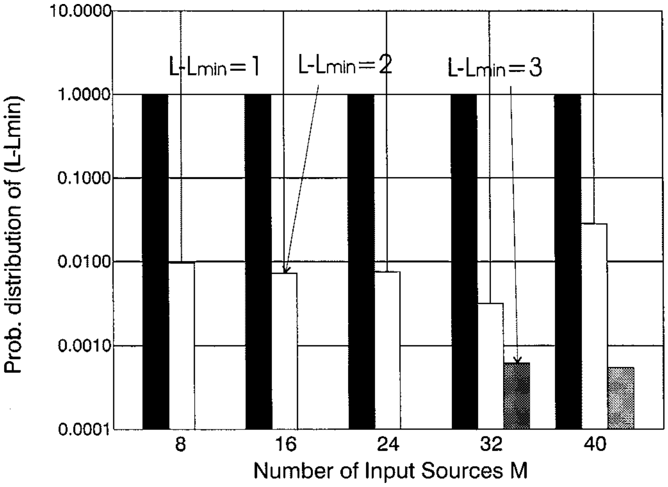

Fig. 8. Probability distribution of the frame length difference between the three-phase algorithm and the optimal.

the percentage increase in frame length as a result of nonoptimal TSAs, against the number of input sources M. We can see that P_F gradually increases with M. However, the percentage increase in frame length remains at a constant value of about 0.1%. That means on the average only one extra time slot is required for a frame length of 1000 time slots.

Fig. 8 shows the distribution of $L - L_{\min}$ for all suboptimal TSAs. When the number of input sources is small, at most two extra time slots are required for any suboptimal assignment. For M up to 40, at most three extra slots are required.

IV. Conclusion

Two efficient TSA algorithms, called the two-phase algorithm for the nonhierarchical and the three-phase algorithm for the hierarchical TDM switching systems, were proposed in this paper. The simple idea behind these two algorithms is to schedule the traffic on the critical lines/trunks of a traffic matrix first. Their time complexities are found to be $O(LN^2)$ and $O(LM^2)$, respectively, where L is the frame length, N is the switch size and M is the number of input/output users connected to a hierarchical TDM switch. Since the proposed algorithm does not use backtracking, pipelined operation of packet transmission and packet scheduling can be carried out. Because of the iterative nature of the proposed algorithms, it is found that hardware implementation for these two algorithms are very simple and efficient. Based on the extensive simulation results obtained, the two proposed algorithms are found to be very efficient. For nonhierarchical TDM switching systems, we found P_F the probability that the two-phase algorithm failed to generate optimal TSAs is about 3×10^{-5} and is independent of switch size. For hierarchical switching systems, the percentage increase in frame length as a result of nonoptimal TSA generated by the three-phase algorithm, is only about 0.1%.

References

[1] T. Inukai, "An efficient SS/TDMA time slot assignment algorithm," *IEEE Trans. Commun.*, vol. COM-27, pp. 1449–1455, Oct. 1979.

[2] C. A. Pomalaza-Raez, "A note on efficient SS/TDMA assignment algorithms," *IEEE Trans. Commun.*, vol. 36, pp. 1078–1082, Sept. 1988.

[3] A. Ganz and Y. Gao, "Efficient algorithms for SS/TDMA scheduling," *IEEE Trans. Commun.*, vol. 40, pp. 1367–1374, Aug. 1992.

[4] K. Y. Eng and A. S. Acampora, "Fundamental conditions governing tdm switching assignments in terrestrial and satellite networks," *IEEE Trans. Commun.*, vol. COM-35, pp. 755–761, July 1987.

[5] R. Jain and G. Sasaki, "Scheduling packet transfers in a class of TDM hierarchical switching systems," in *Proc. Int. Conf. Communications*, 1991, pp. 1559–1563.

[6] N. Funabiki and Y. Takefuji, "A parallel algorithm for time-slot assignment problems in TDM hierarchical switching systems," *IEEE Trans. Commun.*, vol. 42, pp. 2890–2898, Oct. 1994.

[7] Y. W. Leung and T. S. Yum, "A modular multirate video switch—design and dimensioning," *IEEE/ACM Trans. Networking*, pp. 549–557, Dec. 1994.

[8] Y. Leung, "Neural scheduling algorithms for time-multiplex switches," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 1481–1487, Dec. 1994.

[9] H. Duan, J. W. Lockwood, and S. M. Kang, "Matrix unit cell scheduler (mucs) for input-buffered ATM switches," *IEEE Commun. Lett.*, vol. 2, pp. 20–23, Jan. 1998.

[10] T. Weller and B. Hajek, "Scheduling nonuniform traffic in a packet-switching system with small propagation delay," *IEEE/ACM Trans. Networking*, vol. 5, pp. 813–823, Dec. 1997.

[11] A. Mekkittikul and N. McKeown, "A pratical scheduling algorithm to achieve 100% throughput in input-queued switches," in *Proc. IEEE INFOCOM'98*, San Francisco, CA, Mar. 1998.

[12] K. L. Yeung, H. Shi, and N. H. Liu, "Performance analysis of a lookahead scheduling algorithm for input-buffered packet switches," *IEICE Trans. Commun.*, vol. E82-B, no. 8, pp. 1296–1303, Aug. 1999.

[13] C. Kolias and L. Kleinrock, "The odd-even queueing ATM switch: performance evaluation," in *Proc. IEEE Int. Conf. Communications*, 1996, pp. 1674–1679.

[14] V. Yau and K. Pawlikowski, "A conflict-free traffic assignment algorithm using forward planning," in *Proc. IEEE INFOCOM'96*, 1996, pp. 1277–1284.

[15] S. T. Chuang, A. Goel, N. McKeown, and B. Prabhakar, "Matching output queueing with a combined input/output-queued switch," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1030–1039, June 1999.

[16] A. C. Kam and K. Y. Siu, "Linear-complexity algorithms for QoS support in input-queued switches with no speedup," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1040–1056, June 1999.

[17] C. Rose and M. G. Hluchyj, "The performance of random and optimal scheduling in a time-multiplex switch," *IEEE Trans. Commun.*, vol. COM-35, pp. 813–817, Aug. 1987.

[18] S. Chalasani and A. Varma, "An improved time-slot assignment algorithm for TDM hierarchical switching systems," *IEEE Trans. Commun.*, vol. 41, pp. 312–317, Feb. 1993.

[19] S. C. Liew, "Comments on "Fundamental conditions governing TDM switching assignments in terrestrial and satellite networks"," *IEEE Trans. Commun.*, vol. 37, pp. 187–189, Feb. 1989.

[20] M. A. Bonucelli, "A fast time slot assignment algorithm for TDM hierarchical switching systems," *IEEE Trans. Commun.*, vol. 37, pp. 870–874, Aug. 1989.

[21] J. Hopcroft and R. Karp, "An $n^{5/2}$ algorithm for maximum matching in bipartitie graphs," *Soc. Ind. Appl. Math. J. Comp.*, pp. 225–231, 1973.

[22] Y. K. Tham, "On fast algorithms for TDM switching assignments in terrestrial and satellite networks," *IEEE Trans. Commun.*, vol. 43, pp. 2399–2404, Aug. 1995.

[23] Y. Ito, Y. Urano, T. Muratani, and M. Yamaguchi, "Analysis of a switch matrix for an SS/TDMA system," *IEEE Trans. Commun.*, vol. COM-25, pp. 411–419, Mar. 1977.

[24] M. Chen and T. S. Yum, "A conflict-free protocol for optical wdma networks," in *Proc. IEEE GLOBECOM*, 1991, pp. 1276–1281.

Kwan L. Yeung received the B.Eng. and Ph.D. degrees in information engineering from The Chinese University of Hong Kong in 1992 and 1995, respectively.

In July 2000, he joined the Department of Electrical and Electronic Engineering, The University of Hong Kong. Previously, he spent five years in the Department of Electronic Engineering, City University of Hong Kong, as an Assistant Professor. His research interests include personal and mobile communications, high-speed networks, scheduling algorithms, and the next-generation Internet.